Implementation of Generic and Efficient Architecture of Elliptic Curve Cryptography over Various *GF(p)* for Higher Data Security

Kirit V. Patel[1], Mihir V. Shah[2]

[1]*Department of Electronics and Communication,*
L.D. College of Eng., Gujarat Technological University, Ahmedabad, Gujarat, India,
kirit@ldce.ac.in

[2]*Department of Electronics and Communication,*
L.D. College of Eng., Gujarat Technological University, Ahmedabad, Gujarat, India,
mihirec@gmail.com

Abstract*: *Elliptic Curve Cryptography (ECC) has recognized much more attention over the last few years and has time-honored itself among the renowned public key cryptography schemes. The main feature of ECC is that shorter keys can be used as the best option for implementation of public key cryptography in resource-constrained (memory, power, and speed) devices like the Internet of Things (IoT), wireless sensor based applications, etc. The performance of hardware implementation for ECC is affected by basic design elements such as a coordinate system, modular arithmetic algorithms, implementation target, and underlying finite fields. This paper shows the generic structure of the ECC system implementation which allows the different types of designing parameters like elliptic curve, Galois prime finite field GF(p), and input type. The ECC system is analyzed with performance parameters such as required memory, elapsed time, and process complexity on the MATLAB platform. The simulations are carried out on the 8th generation Intel core i7 processor with the specifications of 8 GB RAM, 3.1 GHz, and 64-bit architecture. This analysis helps to design an efficient and high performance architecture of the ECC system on Application Specific Integrated Circuit (ASIC) and Field Programmable Gate Array (FPGA).

***Keywords*: Elliptic Curve Cryptography (ECC), Galois Field (GF), Discrete Logarithm Problem (DLP), Scalar multiplication, Public-Key Cryptography.**

(Article history: Received: 7th September 2020 and accepted 28th December 2020)

I. INTRODUCTION

Elliptic Curve Cryptography (ECC) has emerged as an assured public-key cryptography approach for data security. It is a public-key cryptography system that supports the algebraic structure of elliptic curves over finite fields. ECC is relevant to both the discrete logarithm algorithm and integer factorization families. ECC provides security based on the complexity of solving the Discrete Logarithm Problem (DLP) over a group of points on the elliptic curve [1]. Along with its advantage in terms of bandwidth and performance, it offers a similar security level as it is provided by RSA (Rivest–Shamir–Adleman) or discrete logarithm systems with extensively shorter keys [2].

ECC provides grow of different application area from securing internet protocols to embedded systems in the form of wireless sensor networks, Radio-Frequency Identification(RFID) devices, and smart cards [3]. ECC system's performance is determined by the efficient implementation of the arithmetic operation in the underlying finite field. ECC system that is completely realized in software offers the lowest cost and a high degree of flexibility. Detailed literature regarding the efficient implementation of ECC software can be found in [4]. The software implementation is slower compared to hardware implementation due to impractical in time-constrained environments that need fast processing. Due to this limitation, hardware implementation becomes a more suitable option [5].

Recently all the designs mostly vary in the underlying finite field (GF(2^m) and *GF(p)*). The software and hardware co-approach recognized through an enhanced architecture in the form of a coprocessor. It is referred to as a software and hardware co-design. It is observed for both the underlying finite field. A computational system with multiple heterogeneous elements that have limited capabilities is considered a constrained environment. The limitation arises in terms of storage memory, communication bandwidth, processing time, area, and processing power. The resource-constrained devices are Internet of Things (IoT) nodes, RFID tags, and wireless sensor network motes. In recent times, to provide high-security service to new generation device with limited resources is the greatest challenge in the area of data security. The security algorithms are restricted in utilization by bandwidth, hardware, limited power, high speed in real time application, and limited memory [5]. The real time application requires the same level of security services as conventional applications even though having limited memory and power [6].

The hierarchy of the computations concerned in the implementation of ECC cryptosystems is characterized in four levels of operations as shown in Fig.1.The foundation of the ECC system is defined by the finite field arithmetic or modular operations. It carries the squaring, addition, inversion, and multiplication operations. Point addition and

point doubling are the basic building blocks of ECC. Repeating point addition and point doubling operations are used to perform scalar multiplication. The scalar multiplication concept is used in the ECC cryptosystem encryption and decryption process [7].

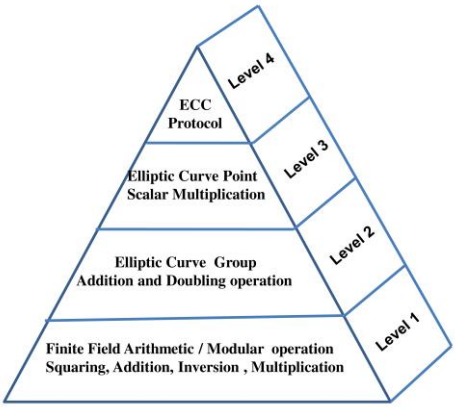

Figure 1: Level of ECC system [2]

The efficient designing of the ECC processor in the hardware platform depends on the various ECC system parameters from the upper layer to the lower layer of the system as shown in Fig.1. Different system parameter provides a challenge to formalize a design structure of ECC in the hardware platform [2].

Section II presents the related work of some important publications related to ECC implementation. Section III presents a short introduction to the theoretical background and inspiration for ECC. Section IV presents an implementation of ECC on MATLAB and the simulation results with considering the different designing parameters. Section V presents conclusion and summary of the simulation results. It also presents a limitation of ECC and a glimpse of future application areas.

II. Related Work

Yan and Chien [8] have implemented an efficient scheme to calculate the finite field multiplication for ECC point. It supports high speed encryption of the message using a dynamic lookup table manner for the decryption and encryption process. The multiplication method is over 70% faster than the standard multiplication by the Russian Peasant method.

Sumit and Brahmjit [9] considered the various important aspects of the ECC system such as finite fields, variety of curve models, and curves together with the attacks on ECC and implemented ECC on the NIST curve p256 with ElGamal encryption. Laiphrakpam and Khumanthem [10] have proposed the algorithm that can do encryption and decryption on type of script with described ASCII values. Srinivasan and Raju [11] have proposed a novel encryption technique of ECC for securing an image that transmits over a public unsecured channel. This technique uses magic matrix operation for providing strong security. Thirumalesu and Sakthivel [12] have proposed a novel hardware architecture for ECC scalar multiplication in Jacobian coordinates on the prime field. It consists of point addition and point doubling architecture. This architecture supports attaining the low hardware resources and high speed using a resource sharing concept which is synthesized both in Field Programmable Gate Array (FPGA) and Application Specific Integrated Circuit (ASIC).

Mainul and Selim [13] have proposed a high-performance modular multiplier. The proposed multiplier has been implemented on Virtex-7 to Virtex-4 series FPGA platforms over the various NIST recommended prime fields p192, p224, p256, p384, and p521. Mainul and Selim [14] have implemented the ECC processor on FPGA. This architecture supports low-area, high-speed, and side-channel attacks resistant ECC processor on a prime field. The processor carries 256-bit point multiplication on the recently suggested twisted edwards curve. Design and implemented multi-key elliptic curve cryptosystem with high-throughput for fast prime field and energy-adaptive dual-field [15]-[18].

III. Basic Concept

The ECC system is designed on the mathematical concepts of elliptic curves and was proposed by Neal Koblitz [19]. An elliptic curve over any field R can be defined as the set of all solutions (x; y) ϵ R x R that fulfill the following general Weierstrass "(1)", where a_i lie in field R.

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_5 \quad (1)$$

Cryptographic application ensures high security because it uses the non singular elliptic curves ($a_1 = 0$; $a_4 = 0$). When R is a finite field, often considered as a Galois Field GF(q), the order of q is equivalent to the number of elements in the Galois finite field. A Galois Field of order q only permits if q is a prime power ($q = p^m$), where m and p indicate positive integer and a prime number respectively. *O* defines the point at infinity and is also measured as a point on the curve. The various finite fields are used for cryptographic applications such as binary fields GF(2^m), prime fields GF(p), and extension fields. Mainly, the use of the extension field is not as frequent as the first two fields. Each field is outfitted with a set of arithmetic operations, mainly defined by additions and multiplications. Elliptic curve multiplication and addition makes the use of underlying finite field arithmetic operation.

2.1 Elliptic Curves

The general Weierstrass equation represents a cubic curve *E* over a field F is shown in "(1)" where a_1, a_2, a_3, a_4, $a_5 \in$ F and the discriminant of *E* is not equal to zero [16]. Alongside this, there is a specified point at infinity which is denoted as *O*. From the general Weierstrass equation, any elliptic curve *E* in its standard form can be written as:

$$E: y^2 = x^3 + ax + b \quad (2)$$

Where the value of *a*, *b* are predefined. Fig. 2 represents $y^2 = x^3 - x + 1$ over the real (R) field when the value of a = −1 and b = 1 are considered. It needs to define binary operation over the elliptic curve because it

satisfies the abelian group. All operations of the abelian group are commutative and some of the operations are shown here.

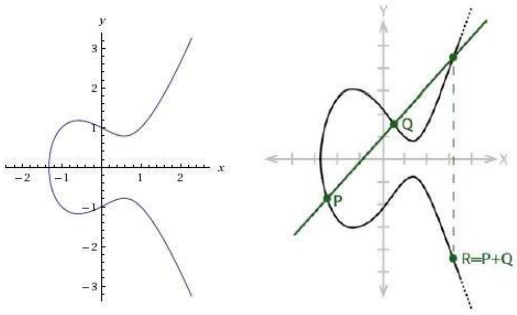

Figure 2: Elliptic curve where a = –1 and b = 1 [15]

Consider P and Q are the two points in the curve. P + Q where P ≠ Q ≠ O outcome in a new point R as (P + Q = R). If P + Q doesn't intersect the elliptic curve, it defines that P + Q is equal to infinity (P + Q = O). This type of outcome happens when P = – Q → P(x, y), Q(x, – y). The results of the operation define the other point if P = O or Q = O. As example P =O, then P + Q = O + Q = Q. 2P is represented as P + Q where P = Q ≠ O and 2P is equal to infinity (2P = O) when the y-coordinate is equal 0.It can be defined that P + Q = O + O = O when P = Q = O.

The coordinate points of the elliptic curve are used for the cryptographic system operation on the elliptic curve over the finite field. Elliptic curve equation over a finite field is defined as " (3)".

$$y^2 = \{x^3 + ax + b\} \bmod \{p\} \qquad (3)$$

A certain elliptic curve point operations are used for the cryptographic function which is defined here. All functions are targeted for the GF(p).

2.1.1 Point addition on elliptic curve

The initial generator points of curve are two point P(x_1, y_1) and Q(x_2, y_2) and both are distinct. The P + Q is defined as P + Q = R(x_3, y_3) and it is given by the following calculation.

$x_3 = \{\lambda^2 - x_1 - x_2\} \bmod p$; $y_3 = \{\lambda(x_1 - x_3) - y_1\} \bmod p$

where $\lambda = (y_2 - y_1 / x_2 - x_1) \bmod p$.

2.1.2 Point doubling on elliptic curve

Now considered that the two point P(x_1, y_1) and Q(x_1, y_1) are overlapped. In this case the P + Q is defined as P + Q = R(x_3, y_3) and it is given by the following calculation.

$x_3 = \{\lambda^2 - 2x_1\} \bmod p$; $y_3 = \{\lambda(x_1 - x_3) - y_1\} \bmod p$

where $\lambda = ((3x_1^2 + a) / 2y_1) \bmod p$

2.1.3 Point multiplication on elliptic curve

Let consider that P is any point on the elliptic curve. Repeated additions are used to calculate multiplication operation over P. It is represented as follows.

kP = P + P + P +· · · + k times.

2.1.4 Point at infinity on elliptic curve

The points is said to intersect at infinity denoted by O if $x_1 = x_2$ and $y_1 = y_2 = 0$ or $x_1 = x_2$ and $y_1 = -y_2$ [10].

2.2 Elliptic Curves over GF(p)

The cryptographic system uses integer points instead of real points across the curve. Let consider that E be an elliptic curve and GF(p) be the finite field with P elements. It needs to consider x = 0, 1, . . . , p – 1 for the calculation of all the points in the finite field GF(p). The finite field F_{191} on curve E: $y^2 = x^3+2x+5$ is shown in Fig. 3.

The elliptic curve has asymmetric feature over y coordinate so it is assured that every valid x–coordinates point on the curve can represent y-coordinates in two diverse points. It is defined as a (mod p) and [$p - a$] (mod p) where a is the square root value in modulo p operation. The number of points on the elliptic curve over the finite field is computed and an estimation of the number of points N is generated.

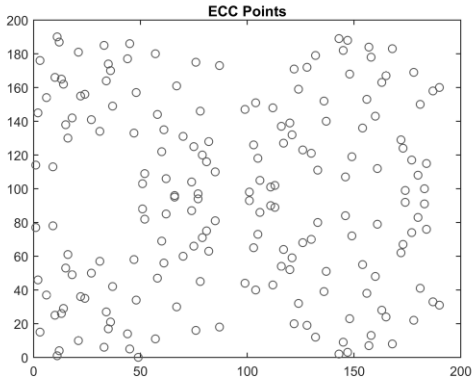

Figure 3: Elliptic curve point on the finite field F_{191}

NIST has suggested the various standard elliptic curves and finite field which can be used for implementing ECC efficiently. The GF(233) and GF(521) are used for the simulation of the ECC system here.

IV. Implementation And analysis

We have integrated different modules of the Elliptic Curve Cryptosystem on MATLAB platform to study the mathematical operation and tried to find out the effective variable to improve the performance of the system. The variable of the cryptosystem plays a major role in performance parameters like memory requirement, speed of the encryption and decryption, type of the input, and complexity of the system.

The generic structure of the ECC is implemented which allows the selection of various types of the elliptic curve, a different type of input, and various types of Galois prime field *GF(p).* This cryptosystem has forced different possible variables and analyzes the cryptosystem It also observes the pattern of encryption point on the elliptic curve and the position of each point concerning the XY coordinate system.

This generic structure has selected the different Galois prime field and compared the elapsed time of encryption and decryption and also compared the memory consumption of specific input and prime field. Based on the simulation results we have proposed an efficient Elliptic Curve Cryptosystem.

Initially, The cryptosystem is implemented using the elliptic curve shown in "(3)". This curve has the characteristics as shown in Fig. 4. The encryption and decryption process on this curve is executed. This curve follows all the required characteristics of ECC such as symmetric about the axis and third point available on the curve. Also, It follows the discrete logarithm problem which provides stronger security features.

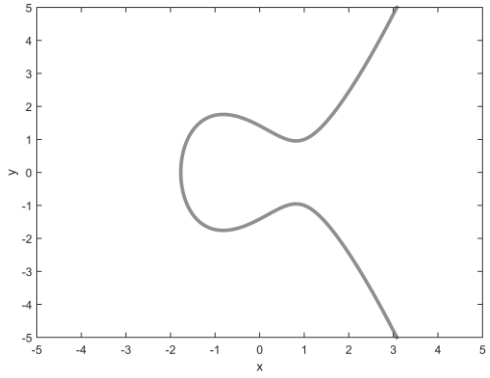

Figure 4: Elliptic curve $Y^2 = X^3 + 7$

Initially, the cryptosystem has selected the Galois prime field GF(233) and generated the possible point on the elliptic curve $Y^2 = X^3 + 7$ as shown in Fig. 5. The cryptography needs the finite field so all the mapping and arithmetic operation executes under the modulo operation of Galois prime Field. Some of the points of encryption point are shown in table 1 with reference to the coordinates system. Based on the random number, the private key is selected and generates the public key using the arithmetic modulo operation on the elliptic curve. These possible points are used to map each message ASCII value on the curve. With the consideration of the elliptic curve and generator point of the curve, the shared key means public key is generated. The encryption process encrypts the message with the help of the private key.

Table1: Coordinate position on Elliptic Curve for GF(233)

X - Coordinate position on curve	215	42	150	198	92	191	18
Y - Coordinate position on curve	0	1	85	2	3	53	108

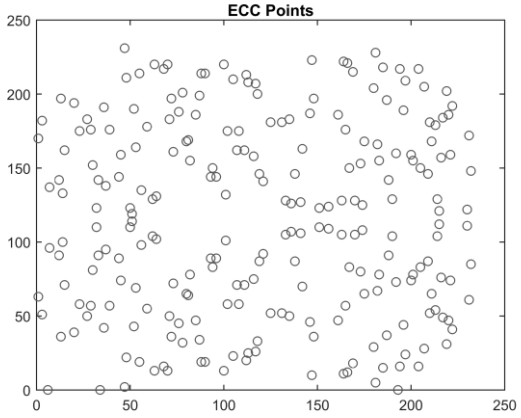

Figure 5: ECC point on the elliptic curve for GF(233)

ECC is an asymmetric type of cryptography which means it uses two different types of keys for encryption and decryption. The public key is used to encrypt the message and the private key is used to decrypt the message. The encryption process executes and encrypts the message with the help of the public key. The mapping of each character of input on the elliptic curve is performed.

The cryptosystem has selected the function parameter such as elliptic curve $Y^2 = X^3 + 7 \bmod p$ where input text size is 1 KB and Galois prime Field is GF(233). The output of the encryption is shown in Fig. 6 in terms of encryption point on the elliptic curve. Encryption point generated after the modular arithmetic operation when public key and the shared key is 212 and 58 respectively in Galois prime field GF(233).

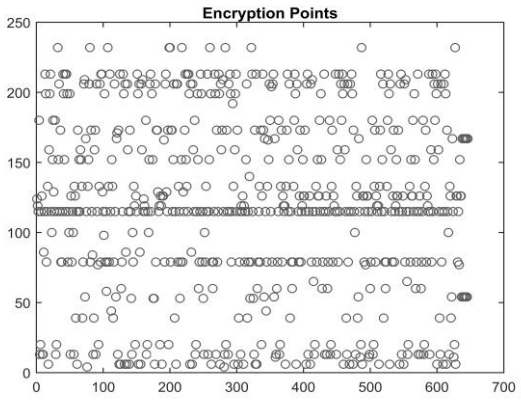

Figure 6: Encryption points on the curve $Y^2 = X^3 + 7$ of GF(233)

The encrypted message is transmitted and the receiver receives the message and decrypts the message with the help of the private key and generates the original message back. Based on the various private key, the elapsed time and memory consumption is compared in table 2. Elapsed time is the average time of 10 times simulation results for the given specific parameters.

Table 2: Function Parameters Comparison for GF(233)

Sr. No.	Random Number (K)	Private Key	Public Key	Elapsed Time (second)	Memory used (KB)
1	124	6	48	0.202361	50
2	31	94	109	0.193078	40
3	3	47	116	0.199403	89
4	39	53	65	0.207299	128
5	104	8	101	0.215078	44
6	2	97	135	0.193301	60

The simulation results show that the cryptographic process uses less memory and less elapsed time when the system is forced with private key 94 and public key 109 for the selected curve and finite prime field. Result clearly shows that there is a tradeoff between elapsed time and memory for the other value of private key and public key.
The input size decides the performance of the cryptosystem. The ECC system is simulated with the increased input size and simulation results are compared in table 3. The effective parameters are forced to the system such as elliptic curve $Y^2= X^3+7$ mod p, input text size 10 KB, and Galois prime field GF(233).

Table 3: Simulation results comparison for Input size 10 KB and GF(233)

Sr. No.	Random Number (K)	Private Key	Public Key	Elapsed Time (second)	Memory used (KB)
1	187	98	176	0.390009	1304
2	214	85	135	0.098616	1444
3	153	192	5	0.180150	184
4	218	170	195	0.100642	488
5	174	152	168	0.094326	610
6	40	132	176	0.098770	1520

The simulation results show that the cryptographic process uses less elapsed time when private key and public key are 152 and 168 respectively. It also shows that the cryptographic process uses less memory when private key and public key are 192 and 5 respectively.

The performance of the cryptosystem depends on the selection of the elliptic curve, so in the second phase, the simulation is carried out on the curve $Y^2 = X^3+ 2X+5$ mod p. The function parameters are selected such as input text size is 1 KB and Galois prime field GF(233). The simulation results are shown in table 4.

Table 4: Simulation results of GF(233) for input size 1 KB

Sr. No.	Random Number (K)	Private Key	Public Key	Elapsed Time (second)	Memory used (KB)
1	136	87	53	0.225200	452
2	199	15	126	0.197295	60
3	120	36	171	0.196724	212
4	56	8	95	0.313080	60
5	56	19	14	0.191335	80
6	211	5	210	0.232228	572

The simulation results clearly suggest that the ECC system gives efficient performance when the private key and public key are 15 and 126 respectively. Now, the system is simulated with an increased input size of 10 KB, and the remaining function parameters are considered the same as previous. The simulation result analysis is shown in table 5.

Table 5: Simulation results of GF(233) for input size 10 KB

Sr. No.	Random Number (K)	Private Key	Public Key	Elapsed Time (second)	Memory used (KB)
1	192	20	200	0.091018	380
2	222	64	42	0.096431	60
3	89	88	150	0.094349	64
4	44	160	175	0.101638	32
5	151	90	199	0.092816	1520
6	65	151	06	0.092652	1260

The comparison result shows that there is a tradeoff between elapsed time and memory for different values of private key and public key. The balance performance is achieved by the value of private key and public key are 160 and 175 respectively. The Galois prime field plays a very important role in the strength of security[16]. The NIST suggests some standard prime field. The cryptosystem has selected GF(521) from the NIST standard and simulated this system. For GF(521) and elliptic curve $Y^2 = X^3 + aX + b$, the possible curve points are mapped as shown in Fig. 7. These all points follow the features of elliptic curve. With increasing points on the curve, the security level is also increased.

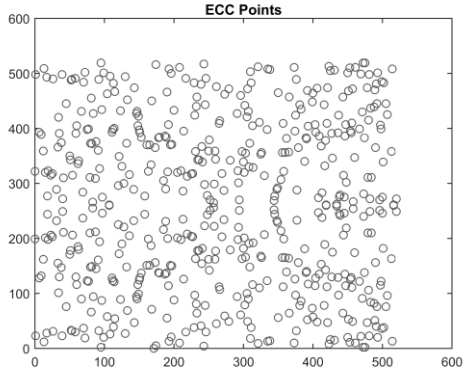

Figure 7: ECC point on the elliptic curve for GF(521)

The output of the encryption is shown in Fig. 8 in terms of encryption point on the elliptic curve. Encryption point generated after the modular arithmetic operation when public key is 178 and the shared key is 458 in Galois prime field GF(521).

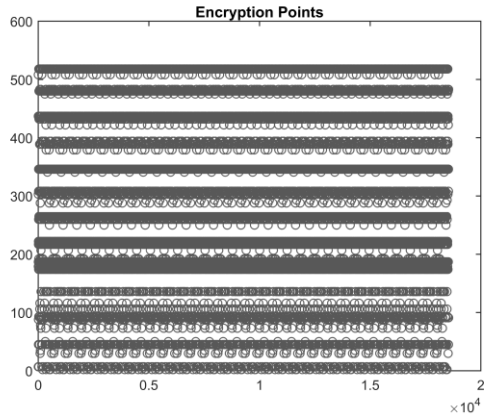

Figure 8: Encryption points on the curve $Y^2 = X^3 + 2X + 5$ of GF(521)

To analyze the ECC system on various standard GF(p), it is simulated for the GF(521) and other functional parameters such as elliptic curve $Y^2 = X^3 + 2X + 5$ mod p and input text Size 10 KB. The simulation analysis is shown in table 6.

Table 6: Simulation results of GF(233) for input size 10 KB

Sr. No.	Random Number (K)	Private Key	Public Key	Elapsed Time (second)	Memory used (KB)
1	500	90	48	0.689574	2808
2	78	14	249	0.210624	504
3	133	85	126	0.290170	440
4	485	25	487	0.130895	292
5	131	20	179	0.144745	740
6	287	59	267	0.232088	5944

For a selected finite field GF(521), the comparison results recommend the value of private key (25) and public key (487) for the balanced performance of the ECC system. To identify the effective parameters of the ECC system for the various GF(p), the simulation results are summarized in a chart. The output of the cryptosystem for the GF(233) and GF(521) are presented in Fig. 9 and Fig. 10 respectively. In all simulations the curve is $Y^2 = X^3 + 2X + 5$ mod p and text size is 10 KB considered.

The Cryptosystem is simulated with the various Galois prime field and simulation results are compared in a chart. The elliptic curve $Y^2 = X^3 + 2X + 5$ mod p and input text size is fixed in this comparison chart.

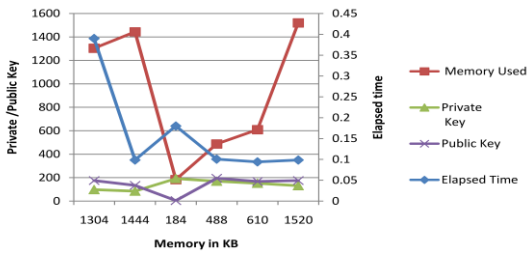

Figure 9: Comparison chart of GF(233) on curve $Y^2 = X^3 + 2X + 5$

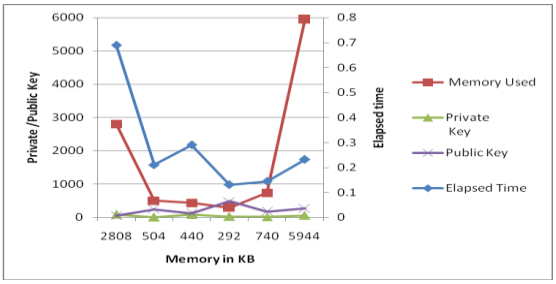

Figure 10: Comparison chart of GF(521) on curve $Y^2 = X^3 + 2X + 5$

The comparison chart shows the tradeoff between speed and memory with the value of private key and public key over the GF(233) and GF(521). From Fig. 10, the system can identify the private key 25 and public key 487 which uses less power and less elapsed time. This comparison result helps to identify effective parameters for hardware implementation such as private key, public key, and type of curve. An effective parameter uses fewer hardware resources and useful in resource constrains applications.

V. Conclusion

The generic and efficient architecture of ECC is presented in this paper. Based on the simulation results, the effective parameters for ECC like Galois prime field, type of curve, and private key can be decided. These effective parameters decide the performance and security level of the ECC system. The effective parameters make efficient and high performance cryptosystem and they help to integrate cryptosystem efficiently on FPGA and ASIC hardware. The proposed architecture supports various GF(p), which provides higher security to the system because of the DLP features. The ECC architecture is limited to third order of elliptic curve. The proposed ECC architecture is suitable for platforms and resource constrains applications like Electronics Commerce, Chip based payment card, Social Media, Digital Currencies, Military Communication, and E-health that require efficiency in terms of area, speed, and power consumption.

VI. References

[1] V.S. Miller, “Use of elliptic curves in cryptography”, Advances in Cryptology CRYPTO 85 Proceedings, Lecture Notes in Computer Science, Springer, Berlin Heidelberg, vol. 218, pp. 417–426, 1986.

[2] H. Marzouqi, M. Al-Qutayri, and K. Salah, “Review of Elliptic Curve Cryptography processor designs,” Elsevier Microprocess. Microsyst., vol. 39, no. 2, pp. 97–112, 2015.

[3] C. Lee and H. Chien, “An Elliptic Curve Cryptography-Based RFID Authentication Securing E-Health System,” International Journal of Distributed Sensor Networks, doi:10.1155/2015/642425, Dec. 2015.

[4] D. Hankerson, J.C. Lopez Hernandez and A.J. Menezes “Software implementation of Elliptic Curve Cryptography over binary fields”. In Cryptographic Hardware and Embedded Systems, CHES vol. 2, pp. 1-24, 2000.

[5] M. Bluhm and S. Gueron, “ Fast software implementation of binary elliptic curve cryptography”, Springer, Journal of Cryptographic Engineering, vol. 5, pp. 215–226, 2015.

[6] C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, "Elliptic Curve Lightweight Cryptography: A Survey," *IEEE Access*, vol. 6, pp. 514–550, 2018, doi: 10.1109/ACCESS.2018.2881444.

[7] I. Setiadi, A. I. Kistijantoro, and A. Miyaji, "Elliptic curve cryptography: Algorithms and implementation analysis over coordinate systems," ICAICT Int. Conf. Adv. Informatics Concepts, Theory Appl., Nov. 2015, doi: 10.1109/ICAICTA.2015.7335349.

[8] Y. Chen and C. Huang, "Efficient Operations In Large Finite Fields For Elliptic Curve Cryptographic" International Journal of Engineering Technologies and Management Research, vol 7, no.6, pp. 141–151, June 2020.

[9] S. S. Dhanda, B. Singh, and P. Jindal, "Demystifying elliptic curve cryptography: Curve selection, implementation and countermeasures to attacks," J. Interdiscip. Math., vol. 23, no. 2, pp. 463–470, 2020.

[10] L. D. Singh and K. M. Singh, "Implementation of Text Encryption using Elliptic Curve Cryptography," Procedia Comput. Sci., vol. 54, pp. 73–82, 2015.

[11] S. Nagaraj, G. S. V. P. Raju, and K. Koteswara Rao, "Image encryption using elliptic curve cryptography and matrix," Procedia Comput. Sci., vol. 48, pp. 276–281, 2015.

[12] T. Kudithi and R. Sakthivel, "An efficient hardware implementation of the elliptic curve cryptographic processor over prime field, Fp," Int. J. Circuit Theory Appl., vol. 48, pp. 1256–1273, 2020, doi: 10.1002/cta.2759.

[13] M. M. Islam, M. S. Hossain, M. D. Shahjalal, M. K. Hasan, and Y. M. Jang, "Area-Time Efficient Hardware Implementation of Modular Multiplication for Elliptic Curve Cryptography," *IEEE Access*, vol. 8, pp. 73898–73906, 2020, doi: 10.1109/ACCESS.2020.2988379.

[14] M. M. Islam, M. S. Hossain, M. K. Hasan, M. Shahjalal, and Y. M. Jang, "FPGA Implementation of High-Speed Area-Efficient Processor for Elliptic Curve Point Multiplication over Prime Field," IEEE Access, vol. 7, pp. 178811–178826, 2019, doi: 10.1109/ACCESS.2019.2958491.

[15] C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, "Lightweight elliptic curve cryptography accelerator for internet of things applications," Ad Hoc Networks, vol. 103, pp. 102159, 2020, doi: 10.1016/j.adhoc.2020.102159.

[16] S. Asif, M. Hossain, and Y. Kong, "High-throughput multi-key elliptic curve cryptosystem based on residue number system," IET Computers and Digital Techniques, vol. 11, no. 5, pp. 165–172, 2017.

[17] S. Gueron and V. Krasnov, "Fast prime field elliptic-curve cryptography with 256-bit primes," J. Cryptograph. Eng., vol. 5, no. 2, pp. 141–151, 2014.

[18] J.Y. Lai, C.T. Huang,"Energy-adaptive dual-field processor for high performance elliptic curve cryptographic applications ", IEEE Trans. Very Large Scale Integr. VLSI Syst., vol.19 (8), pp. 1512–1517, 2011.

[19] N. Koblitz, "Elliptic curve cryptosystems - Mathematics of Computation", Math. Comp. vol 48, pp . 203-209, 1987.

AUTHOR PROFILE

Kirit V. Patel

He is currently working as an assistant professor in the Electronics and Communication (EC) department at L. D. College of Engineering, Gujarat, India. He is pursuing a Ph.D. degree from Gujarat Technological University. He has received MTech. degree in EC with a specialization in VLSI Design from the Institute of Technology, Nirma University in 2009. He has received BE degree from VNSGU, Gujarat in 2006. He has more than 12 years of teaching experience and published 10 research papers in International /National journals/conferences. His main area of research is cryptography and VLSI Front End design.

Dr. Mihir V. Shah

He is currently working as a professor and Head in the Electronics and Communication (EC) department at L. D. College of Engineering, Gujarat, India. He is awarded a Ph.D. degree from MSU, Baroda, Gujarat in 2009. He has received M.E. degree in EC from Malaviya Regional Engineering College Jaipur, Rajasthan in 2001. He has 4 years of industry experience and more than 24 years of teaching experience. He has published more than 30 research papers in International/National Journal / Conference. His main area of research is VLSI Front End design and CMOS analog design.